



Student Name:
Student Number:
Serial Number:

Final Exam, Second Semester: 2015/2016  
Computer Engineering Department

<b>Course Title:</b> Object Oriented Programming	<b>Date:</b> 12/06/2016
<b>Course No:</b> 0630221	<b>Time:</b> 2 hours
<b>Lecturer:</b> Eng. Sultan M. Al-Rushdan	<b>No. of pages:</b> 3

Instructions:

- **ALLOWED:** Non-programmable calculator, pens and drawing tools (**no red colour**).
- **NOT ALLOWED:** Papers, literatures and any handouts. Otherwise, it will lead to the non-approval of your examination.
- **Shut down** Telephones, and other communication devices.

Please note:

- Write your name and your matriculation number on every page of the solution sheets.
- All solutions together with solution methods (explanatory statement) must be inserted in the labelled position on the solution sheets.
- Support your answer with diagrams, equations and examples when possible
- You can submit your exam after the first hour.

**Question 1:**

**(10 marks)**

**Chose the correct answer for the following questions.**

1. Which of the following type of class allows only one object of it to be created?  
A. Virtual class      B. Abstract class      **C. Singleton class**      D. Friend class
2. Which of the following statements is correct?  
A. Base class pointer cannot point to derived class.      **B. Derived class pointer cannot point to base class.**  
C. Pointer to derived class cannot be created.      D. Pointer to base class cannot be created.
3. Which of the following is not the member of class?  
A. Static function      **B. Friend function**      C. const function      D. Virtual function
4. How many instances of an abstract class can be created?  
A. 1      B. 5      C. 13      **D. 0**
5. Which of the following concepts provides facility of using object of one class inside another class?  
A. Encapsulation      B. Abstraction      **C. Composition**      D. Inheritance
6. Which of the following is correct about class and structure?  
A. class can have member functions while structure cannot.  
B. class data members are public by default while that of structure are private.  
C. Pointer to structure or classes cannot be declared.  
**D. class data members are private by default while that of structure are public by default.**
7. Which of the following problem causes an exception?  
A. Missing semicolon in statement in `main()`.      B. A problem in calling function.  
C. A syntax error.      **D. A run-time error.**

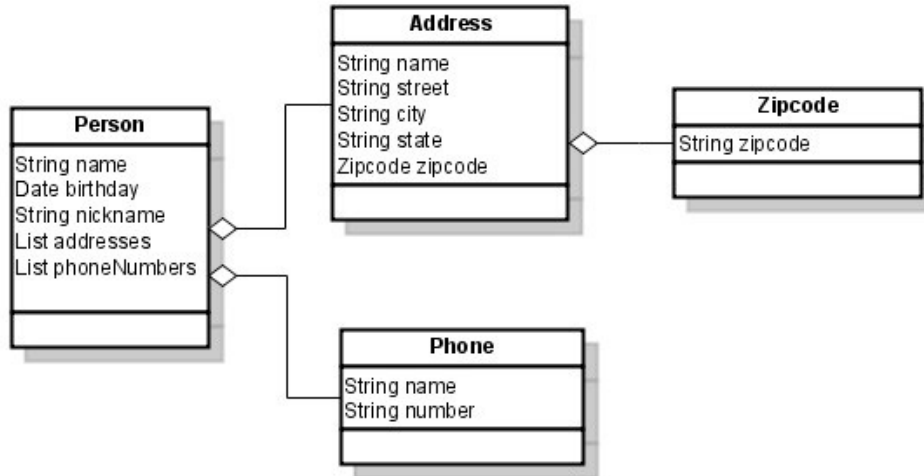


Student Name: Student Number: Serial Number:
--

8. Which of the following is the correct way of declaring a function as constant?  
 A. `const int ShowData(void){/*statements*/}`      B. `int const ShowData(void){/*statements*/}`  
 C. `int ShowData(void) const {/*statements*/}`      D. Both A and B
9. Which one of the following is the correct way to declare a pure virtual function?  
 A. `virtual void Display(void){0};`      B. `virtual void Display = 0;`  
 C. `virtual void Display(void) = 0;`      D. `void Display(void) = 0;`
10. Which of the following statement is correct?  
 A. Class is an instance of object.  
 B. **Object is an instance of a class.**  
 C. Class is an instance of data type.  
 D. Object is an instance of data type.

**Question 2:** **(7 marks)**

Given the following UML diagram construct the code that represents it.



```

class person
{
public:
    person();
    void display();

private:
    string name;
    Date birthday;
    string nickname;
    address adr[10];
    phone phone_numbers[10];
};

class Date
{
public:
    Date();
    void display();

private:
    int day;
    int month;
    int year;
};

class address
{
public:
    address();
    void display();

private:
    string name;
    string street;
    string city;
    string state;
    zipcode z_code;
};
    
```



```
class phone
{
public:
    phone();
    void display();

private:
    string name;
    string number';
};
```

```
class zipcode
{
public:
    zipcode();
    void display();

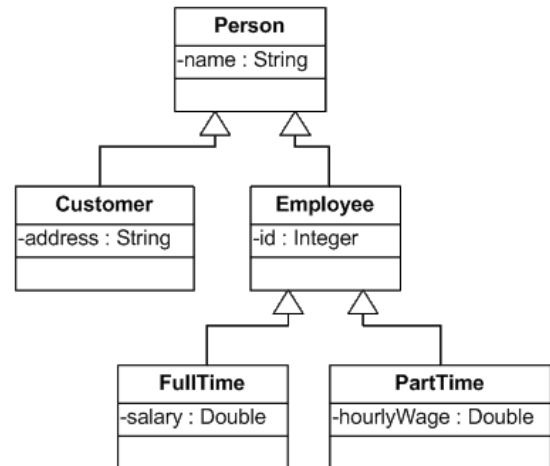
private:
    string z_c;
};
```

**Question 3:**

**(7 marks)**

Given the following class hierarchy construct then code that define the hierarchy below such that

- 1- Person and employee are abstract classes.
- 2- Employee should define PRINT method.
- 3- Employee should define GET\_SALARY method



```
class person
{
public:
    person(string n)
    {
        name=n;
    }
    virtual void display()=0;
protected:
    string name;
};

class employee:public person
{
public:
    employee(string n,int x):person(n)
    {
        ID=x;
    }
    int get_id(){return ID;}
    virtual double get_salary()=0;
protected:
    int ID;
```

```
};

class customer:public person
{
public:
    customer(string n,string
ad):person(n)
    {
        address=ad;
    }
    void display()
    {
        cout<<name<<endl;
        cout<<address<<endl;
    }
protected:
    string address;
};

class full_time:public employee
{
public:
    full_time(string n,int x,double
s):employee(n,x)
```



```

    {
        salary=s;
    }
void display()
{
    cout<<name<<endl;
    cout<<ID<<endl;
    cout<<salary<<endl;
}
double get_salary()
{
    return salary;
}
protected:
    double salary;
};

class part_time:public employee
{
public:
    full_time(string n,int x,double
w,int h):employee(n,x)

```

```

    {
        hourly_wage=w;
        hours=h;
    }
void display()
{
    cout<<name<<endl;
    cout<<ID<<endl;
    cout<<hours<<endl;
    cout<<hourly_wage<<endl;

cout<<hours*hourly_wage<<endl;
}
double get_salary()
{
    return hours*hourly_wage;
}
protected:
    double hourly_wage=w;
    int hours;
};

```

**Question 4:**

**(10 marks)**

Given the following class definition which represent a class that handles two dimension matrix, perform the following tasks.

```

class Matrix
{
public:
    Matrix(int r,int c)
    {
        Row=r;
        Col=c;
        M=new int*[Row];
        for(int i=0;i<Row;i++)
            M[i]=new int[Col];
    }
    void Set_Element(int r,int c,int e)
    {
        M[r][c]=e;
    }

private:
    int Row;
    int Col;
    int **M
};

```

- 1- Overload the + operator that performs the addition operation between two matrices as a member method.
- 2- Overload the - operator that performs the subtraction operation between two matrices as a friend function.
- 3- Overload the \* operator that performs the Multiplication operation between two matrices as a member method.
- 4- Overload the << operator to display the Matrix on screen.
- 5- In Addition and Subtraction operations the two matrices should have the same size (Row and Col) and in Multiplication operation the Col of the first matrix should be the same as the Row of the second matrix, if these conditions are not met the program should throw a class exception of your definition that display a suitable ERROR message and the exception should be passed to the calling function.



```

class Matrix_Exception
{
public:
    Matrix_Exception(string M)
    {
        Message=M;
    }
    string what()
    {
        return Message;
    }
private:
    string Message;
};

class Matrix
{
    friend Matrix operator-
(Matrix&,Matrix&);
    friend ostream& operator<<(ostream&
out,Matrix& t);
public:
    Matrix operator+(Matrix&);
    Matrix operator*(Matrix&);

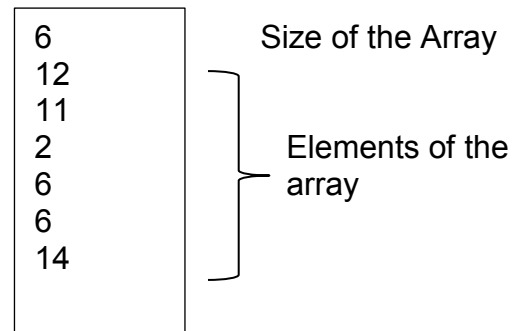
    Matrix(int r,int c)
    {
        Row=r;
        Col=c;
        M=new int*[Row];
        for(int i=0;i<Row;i++)
            M[i]=new int[Col];
    }
    void Set_Element(int r,int c,int e)
    {
        M[r][c]=e;
    }
private:
    int Row;
    int Col;
    int **M;
};
Matrix Matrix::operator +(Matrix &t)
{
    try
    {
        if(Row!=t.Row||Col!=t.Col)
            throw
Matrix_Exception("The matrices should be
in the same size");
        Matrix Temp(Row,Col);
        for(int i=0;i<Row;i++)
            for(int j=0;j<Col;j++)
                Temp.M[i][j]=M[i][j]+t.M[i][j];
        return Temp;
    }
    catch(Matrix_Exception me)
    {
        throw;
    }
};
Matrix Matrix::operator *(Matrix &t)
{
    try
    {
        if(Col!=t.Row)
            throw
Matrix_Exception("The Col of the first
matrix should be the same as the Row of
the second matrix");
        Matrix Temp(Row,t.Col);
        for(int i=0;i<Row;i++)
            for(int j=0;j<t.Col;j++)
                for(int k=0;k<Col;k++)
                    Temp.M[i][j]+=M[i][k]+t.M[k][j];
        return Temp;
    }
    catch(Matrix_Exception me)
    {
        throw;
    }
};
Matrix operator -(Matrix &t1,Matrix &t2)
{
    try
    {
        if(t1.Row!=t2.Row||t1.Col!=t2.Col)
            throw Matrix_Exception("The
matrices should be in the same size");
        Matrix Temp(t1.Row,t1.Col);
        for(int i=0;i<t1.Row;i++)
            for(int j=0;j<t1.Col;j++)
                Temp.M[i][j]=t1.M[i][j]-t2.M[i][j];
        return Temp;
    }
    catch(Matrix_Exception me)
    {
        throw;
    }
};
ostream& operator<<(ostream& out,Matrix&
t)
{
    for(int i=0;i<t.Row;i++)
    {
        for(int j=0;j<t.Col;j++)
            out<<t.M[i][j]<<"\t";
        out<<endl;
    }
}

```

**Question 5:**

**(6marks)**

The following figure represent the architecture of a sequential access file which store an array where the first entry in the file represent the size of the array and the rest of entries represent the elements of the array. Write a program that read the file and store the elements of the array in a dynamic allocated array.



```
#include<iostream>
#include<fstream>
int main()
{
    int size;
    int* Arr;
    ifstream file("d:\\data.txt",ios::in);
    file>>size;
    Arr=new int[size];
    for(int i=0;i<size;i++)
    {
        file>>Arr[i];
    }
    cout<<"size="<<size<<endl;
    for(int i=0;i<0;i++)
        cout<<"Arr["<<i<<"]="<<Arr[i]<<endl;

    file.close();

    return 0;
}
```

**Good Luck**

*Eng. Sultan M. Al-Rushdan*